Image Coding Using Orthogonal Basis Functions

November 2004

Oliver Hunt
ojh16@student.canterbury.ac.nz
Department of Computer Science and Software Engineering
University of Canterbury, Christchurch, New Zealand

Supervisor: R. Mukundan
mukundan@canterbury.ac.nz

Abstract

The transform properties of several orthogonal basis functions are analysed in detail in this report, and their performance compared using a set of grayscale test images, containing both natural and artificial scenes. Well-defined image quality measures are used to determine the type of images that are most suitable for compression for a given basis function. The particular transforms that we have examined are the Discrete Cosine Transform, Discrete Tchebichef Transform, Walsh-Hadamard Transform and Haar Transforms.

We have found that the Discrete Cosine Transform and Discrete Tchebichef Transform provide the greatest energy compactness for images containing natural scenes. For images with significant inter-pixel variations we have found that the Discrete Tchebichef Transform and Haar Transform provide the best performance. The Walsh-Hadamard Transform proved to be significantly less effective than either the Discrete Cosine or Discrete Tchebichef Transforms.

Keywords: Discrete Orthogonal Functions, Discrete Tchebichef Transform, Image Reconstruction, Image Compression.

Acknowledgements

I would like to thank Mukundan for his invaluable assistance during the course of this project, and for his advice when producing the published version of this paper. I would also like to thank Lara Rennie for proofreading a number of the drafts leading up to this final report.

CONTENTS

1 Introduction

The amount of data stored electronically has been increasing rapidly since the dawn of the computer era. The problem of handling this data has become one of the longest lasting problems in computer science, as it becomes increasingly necessary to reduce the amount of physical memory required to store such data. A significant number of compression algorithms have been developed to do this, albeit with varying degrees of success when processing textual information.

As computers became more widespread, images started to be stored on computers. Even small images can require large amounts of storage, meaning compression algorithms rapidly started to demonstrate weaknesses. The compression algorithms that were initially used for processing images were based on the early text compression algorithms. This meant that they were entirely 'lossless' – the decompressed image is identical to the input image. Because they were lossless, the maximum compression rates that could be achieved were limited by the apparent entropy of the image[19]. Given the general approach taken by these older algorithms the overall performance was somewhat lacklustre. Many studies were hence carried out on how to perform image compression. Initially research was focused on lossless techniques, which we describe briefly in Section 2.1. However, these were still of limited ability.

It was then realised that, while decompressed text needed to be identical to the original source, a decompressed image does not need to be, as the image can still be recognised even if it had changed slightly. By using a compression algorithm that did not guarantee identical output, Shannon's limit no longer applied. These 'lossy' compression techniques have become the mainstay of media compression with uses ranging from images to audio and video.

Lossy techniques work by transforming the colour information from an image into some other domain in which information is more compact. A number of lossy compression algorithms do exist, briefly described in Section 2.2. Of interest to us is a particular type of lossy compression system that is based on the use of orthogonal moments.

1.1 Summary

Orthogonal moments have demonstrated many desirable properties in the field of image processing, especially in feature and object recognition. However they also demonstrate significant energy compaction properties. In this paper we present the results of a study of the data compaction properties of discrete transforms of the following form:

$$\tau(u,v) = \sum_{x}^{N-1} \sum_{y}^{N-1} f(x,y)g(x,y,u,v) \tag{1.1}$$

u and v are coordinates in the transform domain, $f(x,y)$ is a function returning the intensity of an image at coordinates x and y, and $g(x,y,u,v)$ is the kernel function for the transform.

By using different moment sets to provide the kernel, g, it is possible to compare the energy compactness of the resultant data sets. In this study we have used the Discrete Cosine Transform (DCT), the Discrete Tchebichef Transform (DTT), the Haar Transform, and the Walsh-Hadamard Transform (WHT). While numerous other moment sets exist ranging from the trivial identity transform through to the more complex (such as the Discrete Fourier Transform), we have limited our study to just these four transforms. Our rationale for this will be discussed, along with more detailed descriptions of the transforms in Section 4.2.

By analysing the Peak Signal to Noise Ratio of the reconstructed image as more components of the transformed data are used, we have discovered that the DTT performs very similarly to the DCT, performing only marginally worse on photographic images, and often much better in images demonstrating rapid gradient changes (typically present in "vector-art"), although in such images the Haar Transform also performed very well. While the WHT showed an improvement in its energy compaction properties when processing vector art, its performance was significantly below that of the other three transforms. With the exception of images that demonstrate large inter-pixel variation the Haar transform typically provide a degree of energy compaction that places it below the DCT and DTT, but still ahead of the WHT.

1.2 Outline

A brief introduction to the concepts and reasons for image compression have been given in Chapter 1. A more in depth look at the two major approaches taken by image compression algorithms is given in Chapter 2. Following that brief exposé Chapter 3 will discuss the motivation for this project, and state our objectives.

The derivation and formal definition of the compression process, as well as computational aspects, are given in Chapter 4. This chapter also provides a detailed description of each of the basis functions that we have analysed.

The results of our analysis, and our research methodology are discussed in Chapter 5.

Chapter 6 describes the limitations of our research, and presents a number of potential areas for further research.

The final chapter of this report contains a summary of the research presented.

2 Background

The problem of information management is not new, especially that of managing of storage and bandwidth requirements. The field of data compression has hence been extensively researched. One of the earliest studies on the theory of data compression was produced by Shannon in 1948[19]. In this paper Shannon stated the limits of data compression, saying what could, and what could not, be done. However, the limits set by Shannon only apply to lossless compression systems. We will discuss some of the lossless systems that have been used for image compression in Section 2.1.

Even small images typically require a large amount of storage space. The limits forced by lossless compression systems hence mean that even compressed images required significant disk space. Unlike textual data, the quality of an image is largely subjective, and hence affected by human perception. It is therefore possible to design compression systems that do not guarantee perfect reconstruction of the input image. By doing this, these systems can achieve very high reconstruction accuracy. The systems provide what is called 'lossy' compression, and we discuss them in Section 2.2.

2.1 Lossless Image Compression

Algorithms to perform lossless compression of data have been studied since the earliest computers were available. As an image is only a stream of bits, when reduced to its fundamentals, all those algorithms developed for compressing general data can be applied to images as well. However systems such as the Lempel-Ziv algorithms[25, 29, 30], and Run-Length Encoding (RLE) rely on pieces of the input stream repeating frequently, a feature common in diagrams, but often lacking in natural pictures. Probabilistic techniques such as arithmetic and Huffman coding[10] can be more effective when processing images as images often exhibit some form of intensity imbalance. Although in order to improve the compression rates for images better predictive models are needed.

Unlike other data forms, the value of one pixel in an image is highly correlated with the intensities of the surrounding pixels. This led rapidly to the realisation that the first derivative of an image often contained significantly less information than the source image itself[4]. A number of primitive image compression techniques were hence developed, such as delta modulation[18] and delta pulse code modulation (DPCM). In these techniques the differences between consecutive pixels are encoded[1], rather than the values of the pixels themselves. If we examine Figure 2.1 we can see the difference in distribution of values in the source image, compared to

[1] This is the first derivative of the image, as it represents the gradient of pixel intensities in the image. Given images are discrete it is trivial to show that the differences between consecutive pixels is the derivative of the image.

the distribution of the differences. Such extreme variations in frequency are ideal for compression as they demonstrate significant redundancy. By compressing this 'derivative' image much greater levels of compression were obtained. This approach is used at some stage by almost all image compression systems – The Graphics Interchange Format (GIF) for instance uses only this approach, coding the output directly with the Lempel-Ziv-Welch algorithm[25].

Due to the increasing size of images, further compression was still needed. Research now started analysing the use of two-dimensional transforms. The best example of this comes from the development of the blockwise transforms that we are studying. Though typically used as the basis of lossy compression systems (see Section 2.2), a number of compression systems were developed that also made use of blockwise image processing.

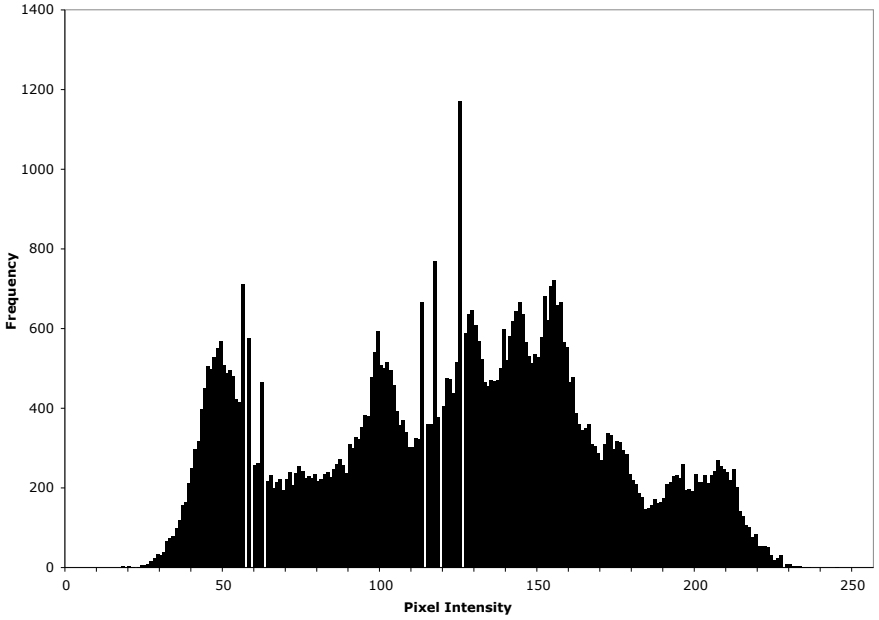
The lossless blockwise compression systems originated with older systems such as block-encoding. Block-encoding was designed to function on grayscale images, in which each block was given a unique index. The image could then be compressed by storing the indexes for each block, rather than the blocks themselves. Compression occurs so long as the image does not containing every possible block (for a 256 level image this would require an image contain $256^{N \times M}$ pixels, assuming each block was $N \times M$ pixels in size). This form of compression has significant overhead as it is necessary to transmit the block index as well as the compressed image. It is therefore possible for the total size of the compressed image combined with its index to exceed the size of the original.

A more effective and more common technique is to use the image to provide a two dimensional context for each pixel. This approach allows contextual information to be applied to the prediction process for each pixel. Delta modulation and DPCM are effectively systems that treat the image as a one dimensional data stream, using only the previously coded pixel to predicate the correct value of the next pixel. Once the pixel is predicted then the correction required to convert the predicted value to the actual value is coded and stored. In the case of delta modulation and DPCM we are effectively predicting that the next pixel will be the same as the one we have just processed.

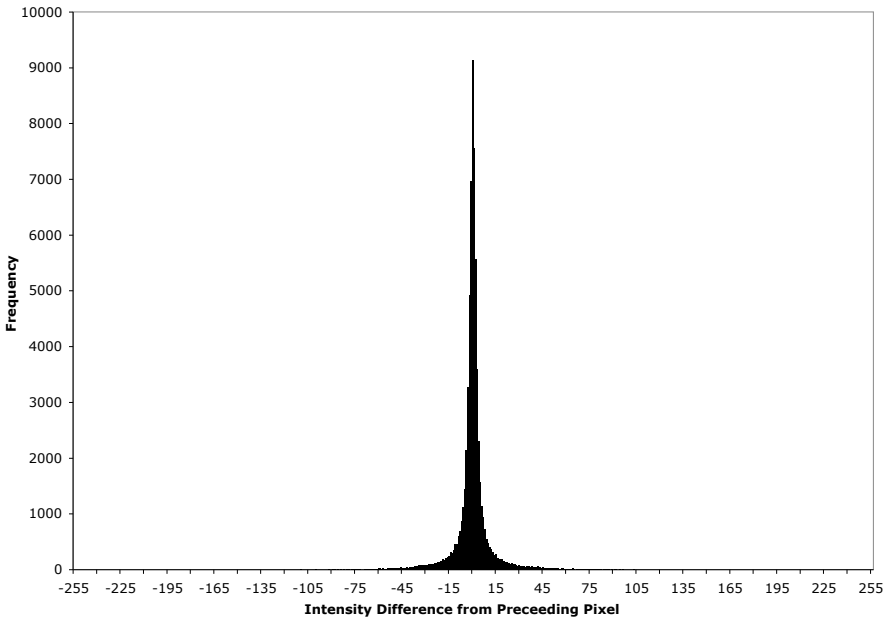
To improve the predictive abilities of the coder, and thus the rate of compression, more than one preceding pixel can be used for prediction. However, as more pixels are used the performance tends to degenerate as the algorithm is degenerating to a standard adaptive coder. The performance can instead be further improved by lowering the weight given to farther away pixels[26].

The best performing lossless image compression algorithms currently available are based on this technique. One of these, although uncommonly used is the CALIC (Context-based Adaptive Lossless Image Codec)[13, 26, 27, 28]. In this system 12 pixels are used to predict the value of the next. To predict the next value four estimates of the local gradient are calculated (across the horizontal and vertical axis, as well as both diagonals). The algorithm then examines previous pixels in order to find possible edges, and adjusts its prediction accordingly. The final correction to the prediction comes through the analysis of the nearest six pixels. These are used to adjust for potential patterns in the input stream. At this point the error between the predicted value and the actual value of the pixel to be coded is stored. This system has proved to be one of the highest performing lossless compression algorithms currently available, achieving rates of only slightly above 3 bits per pixel (3.06bpp) on average, and less than one bit per pixel in some images[13].

The CALIC algorithm is very effective at compressing images, but it is also very good at consuming processing time, as the algorithm is very complex. The JPEG-LS or LOCO-I system[23, 24] is much faster than the CALIC system, yet provide a similar level of performance, also achieving an average compression rate of nearly



(a) Histogram of Pixel Intensity Distribution



(b) Histogram of Intensity Difference Distribution

Figure 2.1 Intensity and intensity difference histograms for the 'Lenna' image. As we can see the intensity difference distribution is much more compact than that of the original image (standard deviation for the intensity image is 47.77, vs 11.69 for the difference image), this greatly reduces the entropy of the image being coded. This significantly improves the achievable compression rates.

3 bits per pixel(3.18bpp)[13]. Unlike CALIC, JPEG-LS uses only four preceding pixels to predict the next pixel. However, just as in CALIC these preceding pixels are used to calculate the local gradient of the image. The significant performance improvement comes from a much simpler edge detector. The edge detector for JPEG-LS requires only three of the pixels, and the prediction function itself is trivial[26]. As with CALIC the coder then attempts to match the current context with special cases of which it is already aware. Given the reduced context size this is also less complex than in CALIC. At this point the final predicted value for the next pixel has been calculated, and the error between the predicted and actual value can be coded and stored.

While these algorithms provide exceptionally good performance for lossless compression, for many applications the resultant images are still too large. Unlike text, images can be distorted slightly yet still be perceived as being similar (or even identical) to the original image. This means that an image compression system need not produce identical output to still be usable. Such system are referred to as being 'lossy' and are discussed in Section 2.2. The lossy systems provide much greater levels of compression, with most easily outperforming all of the lossless image compression systems.

2.2 Lossy coding

Unlike other data sources, such as text, or numerical information, in which any errors are obvious the human eye can compensate for some distortion in images. For this reason it is possible to create compression algorithms that do not need to guarantee that the compressed image will match the original image. By allowing such 'lossy' compression the performance of such a compression is no longer capped by the Shannon limit. This means that it becomes possible for lossy compression systems to achieve extremely high compression rates (around one bit per pixel, or less)[26].

2.2.1 Performance Analysis

It is more difficult to qualify the performance of lossy compression systems than it is for lossless systems. For example, one lossy compression algorithm might merely store the average intensity of the image – leading to a reconstructed image that is just the average of the source. Such a system would provide a very high level of compression, but would be completely useless as the error level would be so high. This means the level of error is another important measure of performance in lossy systems (although most modern lossy compression systems allows the user or developer to trade off quality for compression performance). The most common measures for error are the Mean Squared Error (MSE) and Peak Signal to Noise Ratio (PSNR). These provide a guide to the actual reconstruction accuracy of the algorithm. Even this is not enough to fully qualify the performance of the algorithm, as different types of error are *perceived* differently.

As the error in an image is perceived rather than measured directly by a viewer it is difficult to find someway of rating the quality of an image in a way that allows objective analysis. In order to solve this problem the Picture Quality Scale (PQS) was developed[2]. The PQS provides an objective measure that can be used to quantify the quality of a reconstructed image, taking into account many of the features of the Human Visual System. This allows developers to better judge the usefulness and performance of a compression system.

2.2.2 Transform Based Image Compression

A number of lossy compression systems currently exist. In the trivial case, there are image formats such as GIF that place a low limit on the number of different colours in an image. These require quantisation of colours, and hence are lossy. However, in such cases the loss is caused by limitations of the format, rather than any inherent weakness in the algorithm itself.

Most of the high performance lossy image compression algorithms are currently based on the use of two-dimensional image transforms. The goal of these transforms is to convert image data from the spatial domain to some other domain in which the image exhibits more readily reducible features. The output from these transforms can be converted back to the original image with no loss occurring. However, the significant performance advantage of using these transforms comes from how the image information is contained in the transform domain[22].

Many transforms have the ability to transform the intensity information of an image into frequency related information. The impact of this is twofold. Firstly, the global features of an image are described rapidly by the low frequency components of the transform, while secondly, it is possible to selectively ignore certain components, with minimal effect on the final image[21, 22].

The most simple way of selecting components is to sequentially select each component until a desired reproduction accuracy is reached. As the first components have the most significant impact on the overall image, a very good reproduction can be achieved using relatively few components. However, in selecting components in this way we are likely to have chosen many components that have little or no effect on the final image. In order to improve performance many compression systems, most notably the JPEG standard, hence use quantisation tables to scale the components[21]. Once scaled, those components below a certain value are set to zero. After this quantisation stage each component is sequentially coded and stored, but now all 'zero' components are ignored. This effectively means that unimportant components are no longer used in this reconstruction. Through this approach extremely high levels of compression are achieved[5, 20, 21].

Ignoring any perceptual variance the Karhunen-Loeve Transform is one of the most effective *energy* compaction properties of any transform, being optimal for a number of properties[12]. However, the transform itself is extremely complex, and, since the basis function is derived on a per image basis, not ideally suited to image compression. The transform is still useful as it can be used to estimate the optimal energy compaction rate for any simple transform using an orthogonal basis function[12].

The Direct Cosine Transform is one of the most popular transforms used for lossy compression and has demonstrated itself to be one of these most effective transforms available[21, 22], coming very close to the performance of the Karhunen-Loeve Transform[12]. This performance has seen it adopted in a large number of standards, both for still images (as in the JPEG standard) and for video (the MPEG standards). The Discrete Cosine Transform itself is discussed in greater detail in Section 4.2.1.

Transforms like the Discrete Cosine Transform belong to a class of transforms which use orthogonal basis functions to define a 'kernel'. The coefficients that make up this kernel have a direct impact of the energy compaction properties of the transform as a whole. Numerous other basis functions exist, although the development of new basis functions (that are not merely a scaled or reordered version of an already known function) is not common, it is not an overly rare occurrence. One of the more recent basis functions is the Discrete Tchebichef Transform, recently developed by Mukundan, et al[14, 16].

2.2.3 Classification of Image Features

One of the goals of this study was to find some mechanism to classify images according to features that affect compression. Grgic, et al[9] describe a pair of measures that can be used to quantify properties of an image that affect the performance of compression. These properties, the Spatial Frequency Measure (SFM), and the Spectral Activity Measure (SAM) evaluate various characteristics of an image in both the spatial and frequency domains. For an $N \times N$ pixel image, f, these measures are calculated as shown in equations 2.1 and 2.4.

$$SFM = \sqrt{R^2 + C^2} \tag{2.1}$$

$$R = \sqrt{\frac{1}{N^2} \sum_{j=0}^{N-1} \sum_{k=1}^{N-1} \left(f(j,k) - f(j,k-1)\right)^2} \tag{2.2}$$

$$C = \sqrt{\frac{1}{N^2} \sum_{j=1}^{N-1} \sum_{k=0}^{N-1} \left(f(j,k) - f(j-1,k)\right)^2} \tag{2.3}$$

SAM is defined as being the ratio of the arithmetic and geometric mean of the Discrete Fourier Transform components, F.

$$SAM = \frac{\frac{1}{N^2} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} |F(j,k)|^2}{\left[\prod_{j=0}^{N-1} \prod_{k=0}^{N-1} |F(j,k)|^2\right]^{\frac{1}{N^2}}} \tag{2.4}$$

By using these measures the authors found it was possible to estimate which of the JPEG and JPEG2000 compression systems would work best on a given image.

3 Research Goals

In this portion of our report we will discuss the reasons why we have chosen to study the use of orthogonal basis for image compression and what we hope to find over the course of our analysis. Section 3.1 discusses our motivation and reasoning for undertaking this research, and Section 3.2 describes our objectives.

3.1 Motivations

In the last decade the continual growth of the Internet has led to an extraordinary increase in the amount of information transmitted each every day. Much of this traffic is generated by images of one form or another.

Unlike plain text, images are extremely large when uncompressed, so although the standard text compression algorithms still function on images the compressed file size is still very large. Attempts to solve this problem eventually lead to the development of lossy compression algorithms discussed in Section 2.2. These systems use functions to transform image data into a more readily compressed or reduced form.

The Discrete Tchebichef Transform is a relatively new transform, developed in 2001 (cf. 1974 for the Discrete Cosine Transform) by Mukundan, et al[15, 17]. In studies of the Discrete Tchebichef Transforms it was found that it provided highly effective feature representation capability. Given its ability to represent global features of images effectively it was felt that the Discrete Tchebichef Transform could potentially provide highly effective energy compaction properties when processing images. This supposition was further reinforced by the fact that the Discrete Cosine Transform, one of the most popular transforms used in signal processing today, is also derived from the Tchebichef polynomials. Because of these facts we believed that the Discrete Tchebichef Transform could potentially provide support for significant levels of compression.

The Haar transform is also of interest to us, as the Haar transform is one of the simplest members of a special class of function referred to as a 'wavelet'. These have recently been showing great promise in the field of image compression[5]. The Haar Transform however is very simple, and in fact can be represented in the way described in Section 4.1. Due to its differences from the basis function typically used for this form of compression we felt that it could be interesting to see how well it performed when being used in the same way.

3.2 Objectives

Given our motivations, the most obvious goal of our research is to find out which of the basis functions that we have studied performed better than the rest. Unfortu-

nately the answer to that question is not simple as different basis functions favour different types of image.

Using the techniques described by Exkicioglu, et al[7] and Antonini, et al[3] we will attempt to classify images according to their individual properties. It is our belief that such a classification could be useful in choosing the most effective transform for any given image.

Thus we have two goals, detailed formally below.

- To analyse the performance of the Discrete Cosine Transform, Discrete Tchebichef Transform, Haar Transform and Walsh-Hadamard transform. This analysis would need to cover a variety of different image types and styles.
- To classify the image properties in which the performance of each transform is, better than, similar to, or worse than the other transforms.

4 Image Compression

In this chapter we discuss the algorithm that we have used to perform image compression and the basis functions that we have analysed. Section 4.1 describes the transform itself, and discusses steps that can be taken to improve the performance of the algorithm. The basis functions we have studied are detailed in Section 4.2.

4.1 Discrete Image Transform

As stated in the introduction, our research has been on the analysis of different basis functions for a specific type of image transformation. The image transform we are using is well known, and is the basis of DCT based image compression.

In this Section we will describe the derivation of the algorithm and discuss mechanisms that we have used to improve the algorithmic complexity.

The pointwise definition of the transform we are using for an $N \times N$ pixel image is given in equations 4.1 and 4.2. As the transform is discrete over the domain of the image it must be calculated for $u, v \in [0..N)$.

$$\tau(u,v) = \sum_{x}^{N-1} \sum_{y}^{N-1} f(x,y) g(x,y,u,v) \tag{4.1}$$

$$f'(x,y) = \sum_{u}^{N-1} \sum_{v}^{N-1} \tau(u,v) h(x,y,u,v) \tag{4.2}$$

Equation 4.1 defines the image transform, τ, where u and v represent coordinates in the transform domain, f is the image being transformed, f' is the inverse transform, and $g(x,y,u,v)$ is the basis function used by the transform. Equation 4.2 gives the inverse transform, in this $h(x,y,u,v)$ represents the inverse to the basis function g. As our study is on image analysis g and h are discrete in the image domain.

Section 4.1.1 discusses performance considerations of equation 4.1. Further improvements are then covered as we discuss blockwise computation of the transform in Section 4.1.2.

4.1.1 Algorithmic Complexity

The complexity presented by this transform is therefore $O(N^2 O(g))$ for each element in the output. As the output of the transform has the same number of elements as the input image, the complexity for transforming an entire image is $O(N^4 O(g))$.

The basis function g may be computationally complex as well, however its output is constant, and hence could be cached. A problem with this is found in the size of the cached output. g is a 4 dimensional function, and hence the memory requirements to cache the output become astronomical.

All the basis functions we have analysed were originally used for single dimensional processing. The two dimensional version of these functions can typically be derived as a series of one dimensional functions. Such functions a referred to as being 'separable', we can derive the separable two dimensional function as follows:

$$\tau_1(u) = \sum_{x=0}^{N-1} f(x)g(x,u) \tag{4.3}$$

$f(x)$ is our 1D data source.
substituting $f(x,y)$ for $f(x)$ gives:

$$\tau_1(u,y) = \sum_{x=0}^{N-1} f(x,y)g(x,u) \tag{4.4}$$

The transform been performed across x, now we transform across y:

$$\tau_2(u,v) = \sum_{y=0}^{N-1} \tau(u,y)g(y,v) \tag{4.5}$$

substitute equation 4.4

$$\tau_2(u,v) = \sum_{y=0}^{N-1}\sum_{x=0}^{N-1} f(x,y)g(x,u)g(y,v) \tag{4.6}$$

From this it is possible to see that the any 2 dimensional transform of this form is separable. Therefore we can replace our 2 dimensional transform with the product of two 1 dimensional transforms. Our memory requirements drop from N^4 elements to N^2 elements – the overhead required for the caching the transform output is now lineally associated with the size of the image to be transformed, and is thus much more manageable. With the basis function now cached the algorithmic complexity is now $O(N^2)$ (g is now $O(1)$).

Equation 4.5 hints to a further mechanism to improve performance, once again at a trade off for memory. By processing equation 4.4 and storing the result, we can process equation 4.5 in $O(N)$ time, bringing the total complexity for each point in the transform output to $O(2N)$. This approach leads to equations 4.7 and 4.8 for performing the transform.

$$\mathbf{T}(u,v) = \sum_{i=0}^{N-1} g(u,i)f(i,v), \qquad u,v \in [0..N) \tag{4.7}$$

$$\tau_2(u,v) = \sum_{j=0}^{N-1} \mathbf{T}(u,j)g(j,v), \qquad u,v \in [0..N) \tag{4.8}$$

Now the overall transform complexity is $O(2N^3)$ with memory overhead required for the buffer of kN^2 bytes, where k is the number of bytes of memory required for each element in the basis function or transform output.

We can use an identical approach to reverse the transform. This is due to our use of orthogonal basis functions, meaning the basis function is its own inverse. The process we have described above can be seen as a matrix multiplication:

$$\tau = \mathbf{G}\mathbf{F}\mathbf{G}^T \tag{4.9}$$
$$\mathbf{F} = \mathbf{H}\tau\mathbf{H}^T \tag{4.10}$$

In which $\mathbf{F}$, $\mathbf{G}$, and $\mathbf{H}$ are matrix equivalents of f, g, and h respectively. It is easy to see therefore that the complete process to perform the transform, and then invert it is thus:

$$\mathbf{F} = \mathbf{H}\mathbf{G}\mathbf{F}\mathbf{G}^T\mathbf{H}^T \tag{4.11}$$

In order for the transform to be reversible we need to $\mathbf{H}$ to be the inverse of $\mathbf{G}$ and $\mathbf{H}^T$ to be the inverse to $\mathbf{G}^T$, ie. $\mathbf{H}\mathbf{G} = \mathbf{G}^T\mathbf{H}^T = \mathbf{I}$. Given $\mathbf{G}$ is orthogonal it is trivial to show that this is satisfied when $\mathbf{H} = \mathbf{G}^T$. Given $\mathbf{H}$ is merely the transpose of $\mathbf{G}$ the inverse function for g, $h(x, y, u, v)$, is also separable. Hence the complexity of inverse transform can be reduced to $O(2N^3)$ in exactly the same manner as the original transform.

To perform the reconstruction of a transformed image we need to produce the inverse transform. This transform is also trivial derived from the inverse of the one dimensional transform using the same approach as taken for the transform itself. The inverse itself for an image f, and a transformed image, T, is shown in equation 4.12.

$$f(x, y) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} T(u, v)g(x, u)g(y, v), \qquad x, y \in [0..N) \tag{4.12}$$

The performance of the inverse transform can be improved to $O(2N^3)$ using exactly the same as the original transform.

Further improvements can be made through the use of blockwise processing, as discussed in Section 4.1.2.

4.1.2 Blockwise Processing

Using the techniques discussed in Section 4.1.1 the performance of the transform can be improved significantly, although the transform is still very complex. We also have a much more significant memory overhead now. The approach typically used by DCT based compression systems is to perform the transforms on small blocks of the image, instead of whole image. This approach reduces both the complexity of the algorithm and the memory overhead.

If we perform the transform on $M \times M$ blocks the memory overhead becomes kM^2 for our buffer. When modified to process an $N \times N$ image as a series of $M \times M$ blocks our transform becomes:

$$\tau_{pq}(u, v) = \sum_{x}^{M-1}\sum_{y}^{M-1} f(pM + x, qM + y)g(x, u)g(y, v) \qquad p, q \in \left[0..\frac{N}{M}\right) \tag{4.13}$$

Where τ_{pq} performs transforms the data in the p^{th} column and q^{th} row of the $M \times M$ blocks in the source image. From this it is easy to prove that the complexity of the transform for the entire image has been reduced to $O(2N^2M)$. At this point it is obvious that the fastest approach that could be taken would use a block size of one pixel. Due to the nature of our transform however, this would result in merely a scaling of the intensity of each pixel, and would not result in an increased level of redundancy.

It is apparent that care must be taken when choosing the block dimensions as using a block that is too small reduces the number of output elements that could be quantised. Unfortunately we have problems when we increase the block size. Foremost is the obvious performance difficulties as the complexity will increase from $O(2N^2M)$ to $O(2N^3)$ as M increases to N. Secondly the values calculated for the basis function often have a large dynamic range. This feature can cause significant processing errors when using complex basis functions, especially when they are defined using recurrence relations as is the case for the Discrete Tchebichef Transform.

To make our choice we examined those made by the JPEG, and MPEG standardisation bodies. These bodies defined the well known 'JPEG' still picture compression standard[21], and the 'MPEG', 'MPEG-2', and 'MPEG-4' motion picture standards[20]. All of the compression systems rely on the use of the DCT (discussed in Section 4.2.1) but used different block sizes.

In 1991 when the 'JPEG' standard was formed the processing power required to perform the DCT was an important consideration. It needed to provide a significant level of compression, whilst still being feasible on the computers of the day. The final choice was to use 8×8 pixel blocks, as the computational overhead required to increase the block size was considered.

The 'MPEG' compression standards, however, use 16×16 pixel blocks for compression (although they also perform inter-frame processing as well). Given the 'MPEG' standards are based on more recent comparisons of the performance of different block sizes we have chosen to follow in their stead. Hence all analysis we perform will be based on the use of 16×16 pixel blocks. For further details on the decisions made when performing our analysis see Section 5.1.

4.2 Basis Functions

In this Section we shall discuss the basis functions we have analysed. We have chosen to refer to each basis function by the name that is typically associated with the combined transform. We choose to do this as the basis functions themselves often have no name, and hence are typically referred to using the name of the transform they define. Figure 4.1 contains the graphical representation of each of the basis functions for an 8×8 pixel transform.

In order to more adequately demonstrate the effect of each transform, Figure 4.2 shows the output of each transform when applied to a 256×256 pixel version of the 'Lenna' image (in Figure 5.1(a)) as a single 256×256 pixel block.

4.2.1 Discrete Cosine Transform

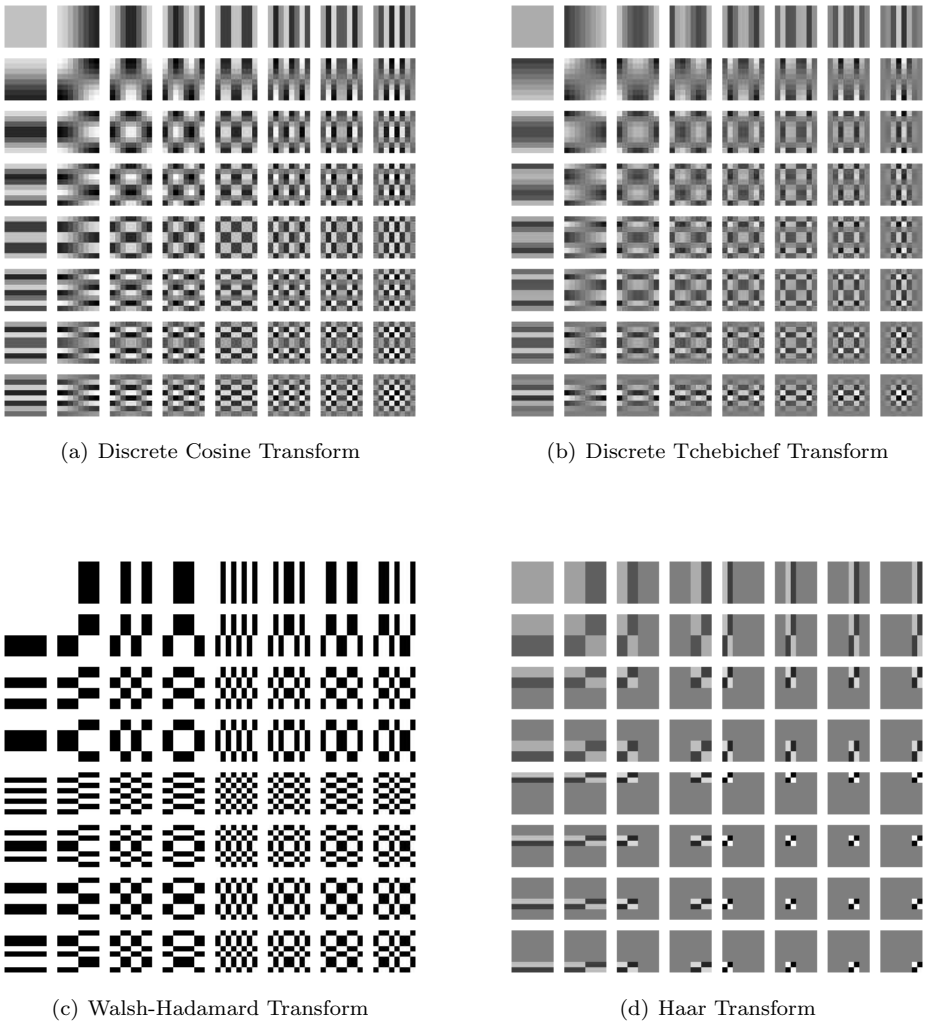
The Discrete Cosine Transform, or DCT, is one of the most well known transforms in image processing, and is used in many different fields, including compression (as the basis for major standards such as JPEG, and MPEG 1, 2 and 4).

The DCT was originally developed in 1974[1], and was first applied to the field of image compression 10 years later, in 1984[22]. This eventually led to the use of the DCT as the basis for the JPEG still image compression standard[21]. Based on its history in the field of image compression the DCT makes an obvious candidate for our research.

The kernel for the DCT is derived from the orthonormal Tchebichef polynomials, resulting in the following definition for the basis function g'[1]:

$$g'(x,u) = \lambda(u)\cos\frac{\pi(2x+1)u}{2N} \tag{4.14}$$

where



(a) Discrete Cosine Transform

(b) Discrete Tchebichef Transform

(c) Walsh-Hadamard Transform

(d) Haar Transform

Figure 4.1 Graphical representation of each of the basis functions studied.

$$\lambda(u) \quad = \quad \begin{cases} \sqrt{\frac{1}{N}}, & u = 0 \\ \sqrt{\frac{2}{N}}, & otherwise \end{cases} \tag{4.15}$$

The final basis function for the DCT is shown in Figure 4.1(a).

4.2.2 Discrete Tchebichef Transform

The Discrete Tchebichef Transform (DTT) is a relatively new transform that uses the Tchebichef moments to provide a basis matrix. As with the DCT the DTT is derived from the orthonormal Tchebichef polynomials, which leads us to presume that it will exhibit similar energy compaction properties[14, 17]. The basis function of the DTT is defined as follows.

$$g'(x,u) = t_u(x) \tag{4.16}$$

In which $t_p(x)$ is the p^{th} order of the Tchebichef moments. These can be defined

using the following function over the discrete range $[0..N)$:

$$t_p(x) = u! \sum_{k=0}^{u} -1^{u-k} \binom{N-1-k}{u-k} \binom{u+k}{u} \binom{x}{k} \tag{4.17}$$

Due to the large dynamic range of the intermediate values generated by equation 4.17 it is not feasible to calculate the values of DTT on a point wise basis. Instead we calculate the function using the following recurrence relation:

$$t_0(x) = \frac{1}{\sqrt{N}}$$

$$t_1(x) = (2x + 1 - N)\sqrt{\frac{3}{N(N^2-1)}}$$

$$t_p(x) = A_1 x + A_2)t_{p-1}(x) + A_3 t_{p-2}(x) \tag{4.18}$$

where $p \in [2..N)$, and A_1, A_2 and A_3 are defined as follows:

$$A_1 = \frac{2}{p}\sqrt{\frac{4p^2-1}{N^2-p^2}},$$

$$A_2 = \frac{1-N}{p}\sqrt{\frac{4p^2-1}{N^2-p^2}},$$

$$A_3 = \frac{p-1}{p}\sqrt{\frac{2p+1}{2p-3}}\sqrt{\frac{N^2-(p-1)^2}{N^2-p^2}}$$

As noted by Mukundan[14] the recurrence relation causes minor numerical errors to propagate through the calculation. This error eventually manifests itself in the collapse of the basis functions. The effect of this collapse is shown in Figure 4.2(b), in which we can see the output from the transform has abruptly become saturated. This problem is only apparent in this image as we are performing the transform over the entire image, rather than on a block-by-block basis. We have found that by performing the transform in blocks of less than 64×64 pixels this problems is safely avoidable, and substantially faster (see 4.1.1).

4.2.3 Haar Transform

The basis function for the Haar Transform is unique among the functions we have examined as it actually defines what is referred to as a 'wavelet'. Wavelet functions are a class of functions in which a 'mother' function is translated and scaled to produce the full set of values required for the full basis set. Of all the known wavelet functions, the Haar wavelet is the oldest and simplest[6]. The Haar wavelet is defined by a simple piecewise function:

$$\Psi\ (x) = \begin{cases} 1, & 0.0 \le x < 0.5 \\ -1, & 0.5 \le x < 1.0 \\ 0, & otherwise \end{cases} \tag{4.19}$$

To use this wavelet it is necessary to scale the wavelet, this is performed using a scaling function, $\phi_{pq}(x)$. For the Haar wavelet this is:

$$\phi_{00}(x) = 1, \qquad x \in [0,1] \tag{4.20}$$

$$\phi_{pq}(x) = 2^{p/2}\Psi(2^p x - q + 1) \qquad x \in [0,1] \tag{4.21}$$

By expanding this we obtain the Haar function,

$$h_{00}(x) = 1, \qquad x \in [0,1] \tag{4.22}$$

$$h_{pq}(x) = \begin{cases} 2^{p/2}, & \frac{q-1}{2^p} \leq x < \frac{q-0.5}{2^p} \\ -2^{p/2}, & \frac{q-0.5}{2^p} \leq x < \frac{q}{2^p} \\ 0, & otherwise \end{cases} \tag{4.23}$$

To obtain the Haar Transform itself we let x assume discrete values a m/N, $m = [0..N)$, hence the basis function for the Haar transform is:

$$H_{km} = h_{p,q}\left(\frac{m}{N}\right), \qquad x, m \in [0..N) \tag{4.24}$$

Where k is uniquely decomposable as

$$k = 2^p + q - 1 \tag{4.25}$$

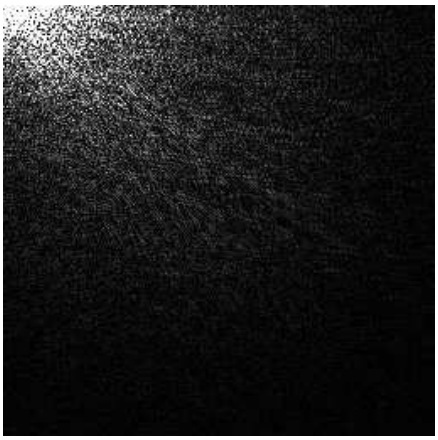
Although we are not using the Discrete Wavelet Transform, as most wavelets would be, the basis function defined by the Haar wavelet is orthogonal, and can therefore be used in this type of transform as well. If we compare the values of the defined basis function, we can immediately see the difference between the basis function of the Haar Transform, in Figure 4.1(d), and the other basis functions. An even more extreme difference is apparent in Figure 4.2(c). Unlike the other transforms we have analysed in which the data in the transform domain is uncorrelated with the original image, the Haar transform space is very clearly correlated. From its structure we can see that the Haar transform "takes the differences of samples or differences of local averages of samples"[12] of the input image.

4.2.4 Walsh-Hadamard Transform

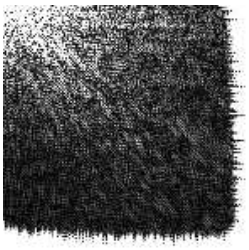
The Walsh-Hadamard Transform (or WHT) is the simplest of the transforms we have studied, and is used primarily as a reference point, allowing us to compare the performance of the complex DCT, DTT and Haar kernels, to a simpler, more readily computed one. The Walsh-Hadamard is the general name given to either the Walsh or the Hadamard Transforms, as they are merely reordering the same components. The kernel is defined as:

$$g'(x,u) = \frac{1}{\sqrt{N}} \prod_{i=0}^{n-1} -1^{b_i(x) * b_{n-1-i}(u)} \tag{4.26}$$

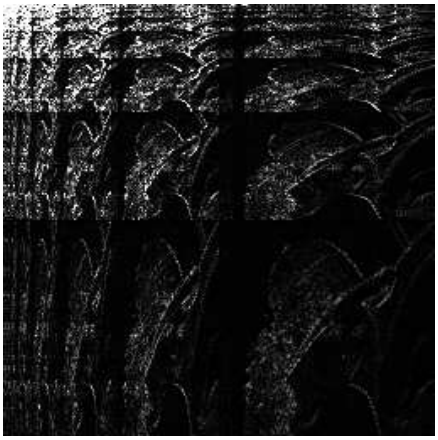
Where $b_i(x)$ returns the i-th bit of x, and $N = 2^n, n \in I$. This results in the basis function shown in Figure 4.1(c). If we examine the output of this transform, in Figure 4.2(d) we can see that the transform contains significant borders between Sections at different powers of 2.



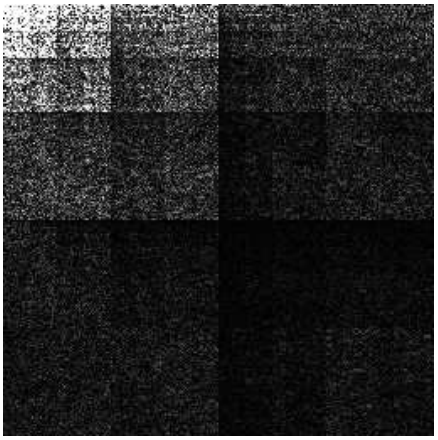
(a) Discrete Cosine Transform



(b) Discrete Tchebichef Transform



(c) Haar Transform



(d) Walsh-Hadamard Transform

Figure 4.2 The 'Lenna' after being transformed using each of the different basis functions. The abrupt termination of the Discrete Tchebichef Transform is caused by the accumulation of error whilst processing the basis function (see Section 4.2.2).

5 Analysis

In this section we will discuss the approach that was taken for comparing the performance of each of the transforms that we have studied. Section 5.1 explains how we performed our study, including how we measured performance, and how we chose to the transform components used to reconstruct an image. Following that, Section 5.2 discusses the results of our study, and how the performance of each transform was affected by the differing image properties.

5.1 Experimental Method

In order to perform a comparison of the energy compactness properties of each of the transforms it was necessary to analyse their performance over a number of different image types. For this reason we have analysed the performance of each transform over a number of standard images from the USC-SIPI[1] image database. The images we have studied include the common 'Lenna' and 'Baboon' images, as well as 'Fishing Boat' and a variety of artificial test images. We have also included the 'Goldhill' image from the Waterloo Repertoire[2] in order to better balance the selection of images we have studied. These images are shown in Figures 5.1 and 5.2. Each is an 8-bit grayscale image, 2^n pixels square.

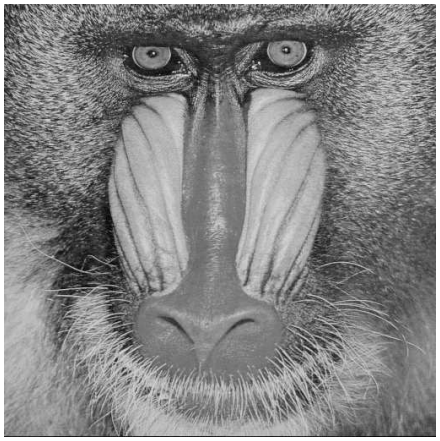
To analyse the energy compactness of each transform it was we measured the reconstruction error on each of the images with when using a different number of components used to perform the reconstruction. All the calculations were performed using double precision floating point values over 16×16 pixel blocks of the image. Given the dimensions of the images we have studied this approach ensured that there were no areas in an image that might be cropped, or otherwise treated differently from others.

In order to classify the properties of images we have used the spatial frequency and spectral activity measures (SFM and SAM). These are defined in equations 2.1 and 2.4. The SFM value acts as a measure of high frequency components in an image. A large value for SFM indicates thats the image contains high frequency information. SAM effectively measures the 'predictability' of an image, it has a dynamic range of $[1..\inf)$. High values of SAM implies high predictability. When performing the SFM and SAM calculations pixel intensities were taken to be within the integral range from 0 to 255 (inclusive).

5.2 Results

[1] `http://sipi.usc.edu/`
[2] `http://links.uwaterloo.ca/bragzone.base.html`

(a) 'Lenna'

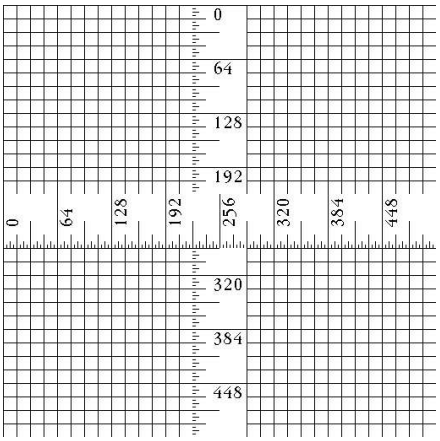


(b) 'Baboon'

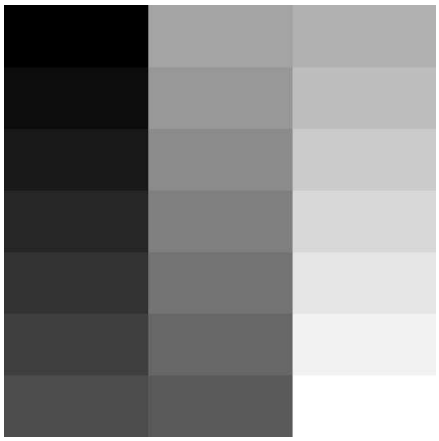
(c) 'Fishing Boat'

(d) 'Goldhill'

Figure 5.1 The first set of images we have analysed consisted of the well known 'Lenna' and 'Baboon' images, as well as the less common 'Fishing boat' image from the USC-SIPI database. We have also included the 'Goldhill' image from the 'Waterloo Repertoire'

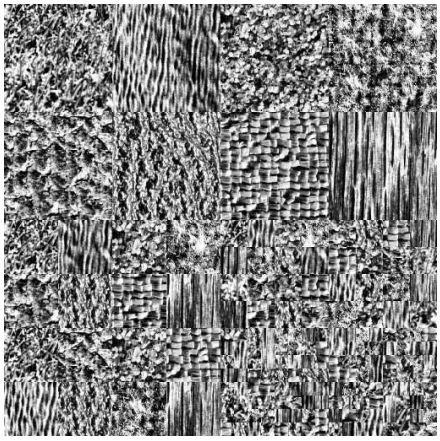


(a) 'Pixel Ruler'



(b) '21 level step wedge'

(c) '256 level test pattern'



(d) 'Texture Mosaic #2'

Figure 5.2 This set of images was chosen to provide an indication of performance when processing images with features uncommon in natural scenes, such as rapid changes in gradient, tonality, or texture.

In this section we will discuss the results gathered from transforming the aforementioned images. We will first discuss measurements taken from each image individually in Section 5.2.1. Section 5.2.2 the results will be summarised, with trends and patterns discussed as well.

5.2.1 MEASUREMENTS

We did not quantise the transform component values as we would if we were actually compressing the data. This means that when all transform components are used to perform the image reconstruction there will be no errors remaining. In the graphs in Figures 5.3 to 5.10 we have included the full set of results for each image. Each graph display the peak signal to noise ratio of the transform, as more of the transform components are used to reconstruct the image. This gives us a way to compare the energy compactness of each transform. The more rapidly the signal to noise ratio increases the higher the energy compactness of the transform for that particular image.

Lenna (Figure 5.1(a)):
SFM: 14.04
SAM: 907.2
Looking at the results in Figure 5.3 we can see that the DCT and DTT provide the highest quality reconstruction, and hence the highest energy compactness. The WHT performs poorly when compared to the other transforms, falling further behind as the number of components used in the reconstruction increases. Initially the performance of the Haar transform matches the performance of the WHT, although once higher order moments are used its performance improves to match that of the DCT and DTT.

Baboon (Figure 5.1(b)):
SFM: 36.55
SAM: 99.61
Given the relatively low predictability of the Baboon image when compared to some of the other images (especially 'Lenna' and the 'Step Wedge') it is not surprising that performance of all of the transforms is lower than in other photographic images. Figure 5.4 shows that the DCT, DTT, and Haar transform all perform nearly identically, the WHT however, falls short of performance of the other three transform as higher order components are used in the reconstruction.

Fishing Boat (Figure 5.1(c)):
SFM: 19.85
SAM: 598.2
In Figure 5.5 we can see that the transforms perform in much the same way for this image as they did for the 'Lenna' and 'Goldhill' images. The only significance is in the performance of the Haar transform. In the 'Lenna' and 'Goldhill' images the Haar transform matches the performance of the DCT and DTT as the higher order components are added. In this image we find that the Haar transform instead performs badly, and matches the performance of the WHT.

Goldhill (Figure 5.1(d)):
SFM: 16.17
SAM: 507.1
For the final of our natural images we studied the 'Goldhill' image, the results of this are shown in Figure 5.6. While this image exhibits slightly less predictable features than the 'Lenna' image the performance of the transforms

relative to each other is more or less identical. The only difference is that the overall performance indicates a lower energy compactness in the transform space for all of the transforms, resulting in a similar reduction in performance for all transform.

Pixel Ruler (Figure 5.2(a)):
SFM: 116.8
SAM: 381.3
In the first of our non-natural images we see a dramatic difference in the performance of the transforms. In this image we see that the DTT clearly outperforms the DCT, and it does so by a significant margin. Despite this it is the performance of the Haar transform that is the most intriguing.

Although initially it performs on par with the DCT, Figure 5.7 shows that the performance of the Haar transform 'spikes' to reach the performance of the DTT. Eventually, it surpasses even the DTT, demonstrating significantly higher energy compactness. The spikes in the performance of the Haar transform might at first be considered directly related to the image, as is the case in the 'Step Wedge' image. However given the in this image the DCT and DTT do not exhibit similar traits we conclude that the spikes are caused by the structure of transform itself. This is further reinforced by similar spikes in the WHT, which shares similar rectangular features to the Haar transform As shown in Figures 4.2(d) and 4.2(c).

21 Level Step Wedge (Figure 5.2(b)):
SFM: 6.539
SAM: 241024
This image is unique among the images that we have studied as it is the only image studied that was completely reconstructed by all the transforms, before all the transform components were used (see the graph in Figure 5.8). This high level of performance is somewhat expected given the level of predictability indicated by the SAM value.

We can also tell that the structure of the image has affected the performance as all of the transforms are showing similar spikes in their reconstruction accuracy. Although the first spike in the Haar transform causes its overall performance to be significantly better than that of the other transforms.

256 Level Test Pattern (Figure 5.2(c)):
SFM: 64.49
SAM: 65.25
The low predictability of this image, coupled with large inter-pixel variation (SFM) results in very poor performance for of all of the transforms. This low performance results in very little variance in the performance of the transform. Figure 5.9 shows that as higher order components get added we start to see some separation in performance, the WHT falls behind, while the DTT and Haar transform lead.

Texture Mosaic #2 (Figure 5.2(d)):
SFM: 75.55
SAM: 30.95
In our final image, we have used a 'texture mosaic'. This is a set of unrelated textures, separated into blocks of differing size. This leads to an image with a very low predictability. Coupled with the high inter-pixel variance indicated by its SFM value, it is not at all surprising to find that all of the transforms performed extremely poorly, this is shown in the graph given in Figure 5.10. Overall the performance of all of the transforms is more or less identical. The

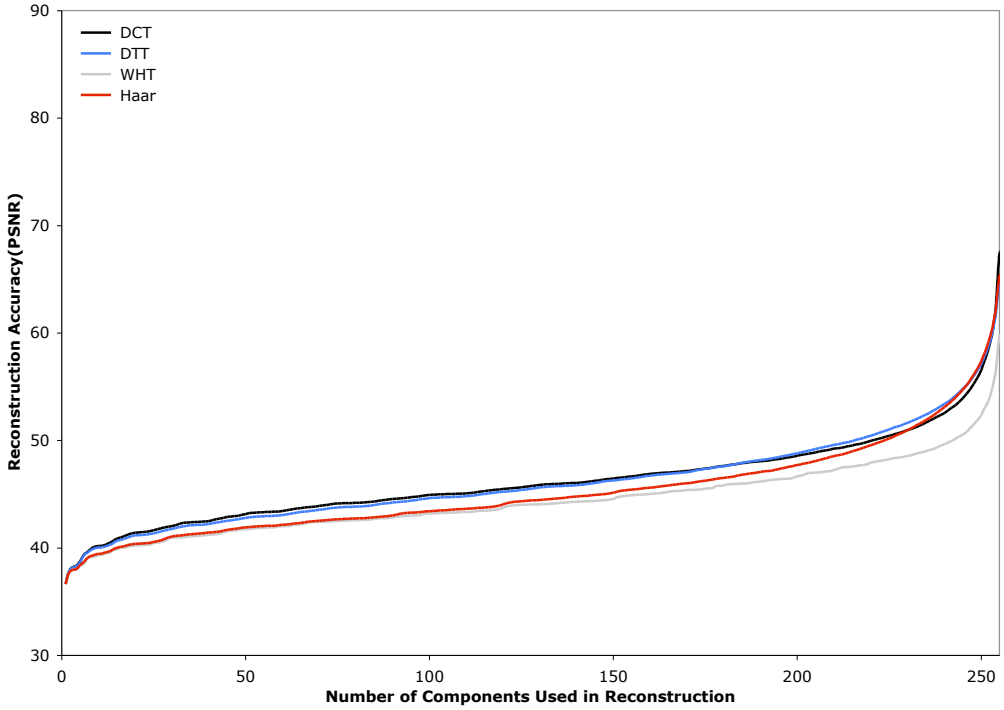


Figure 5.3 Reconstruction accuracy when processing the 'Lenna' image.

only break from this trend is the WHT which has lower energy compactness in the higher order components than the others transforms. Hence the WHT falls behind when processing as the number of components increases.

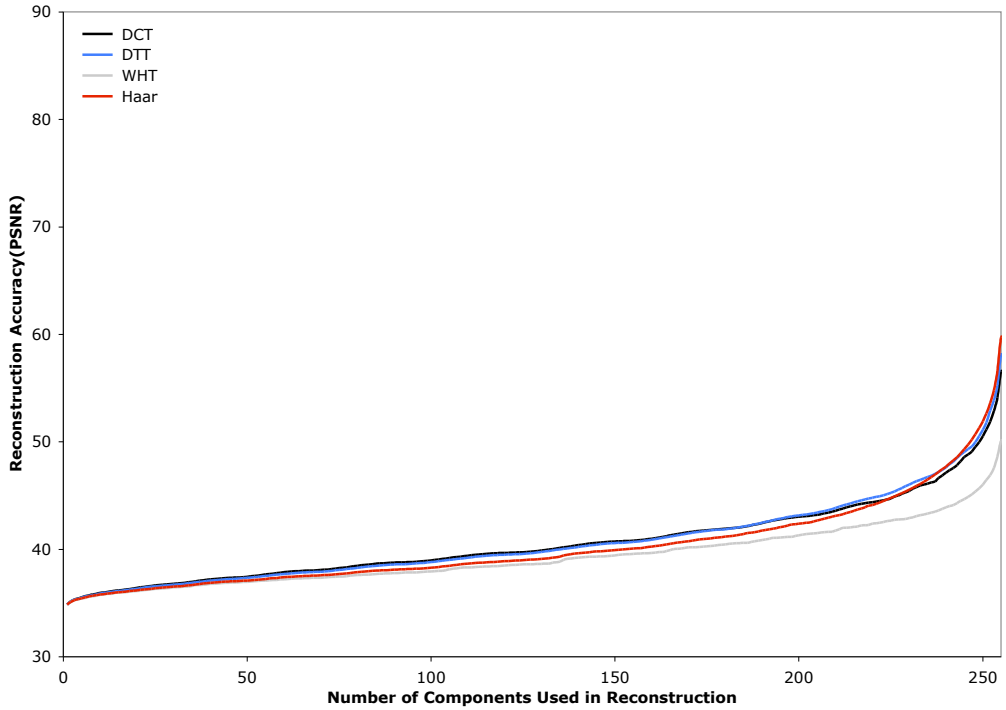


Figure 5.4 Reconstruction accuracy when processing the 'Baboon' image.

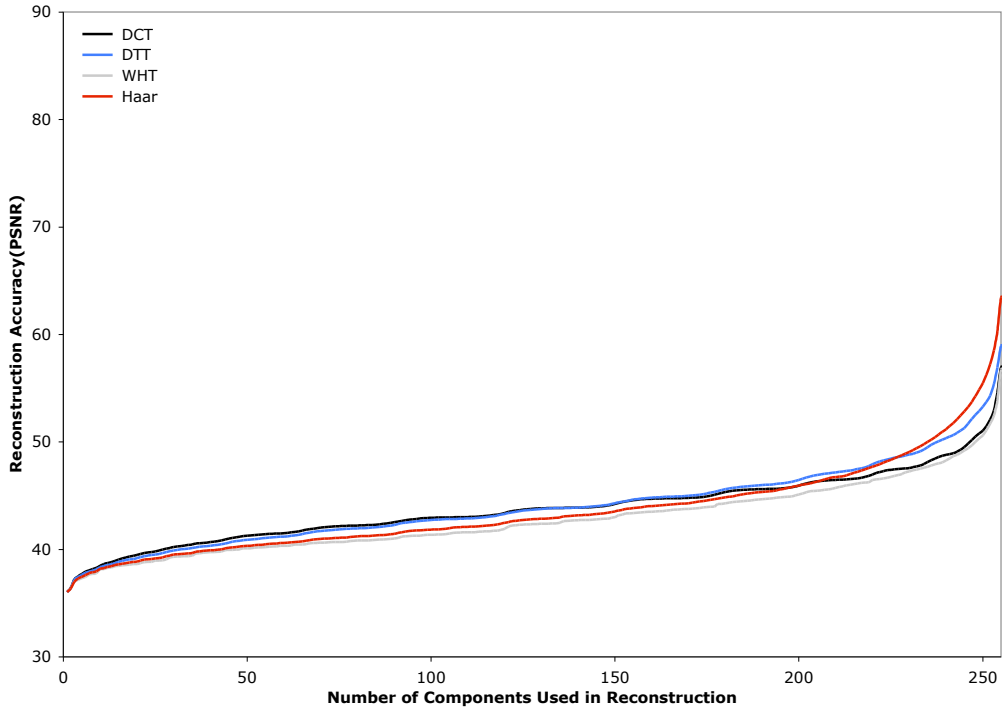


Figure 5.5 Reconstruction accuracy when processing the 'Fishing Boat' image.

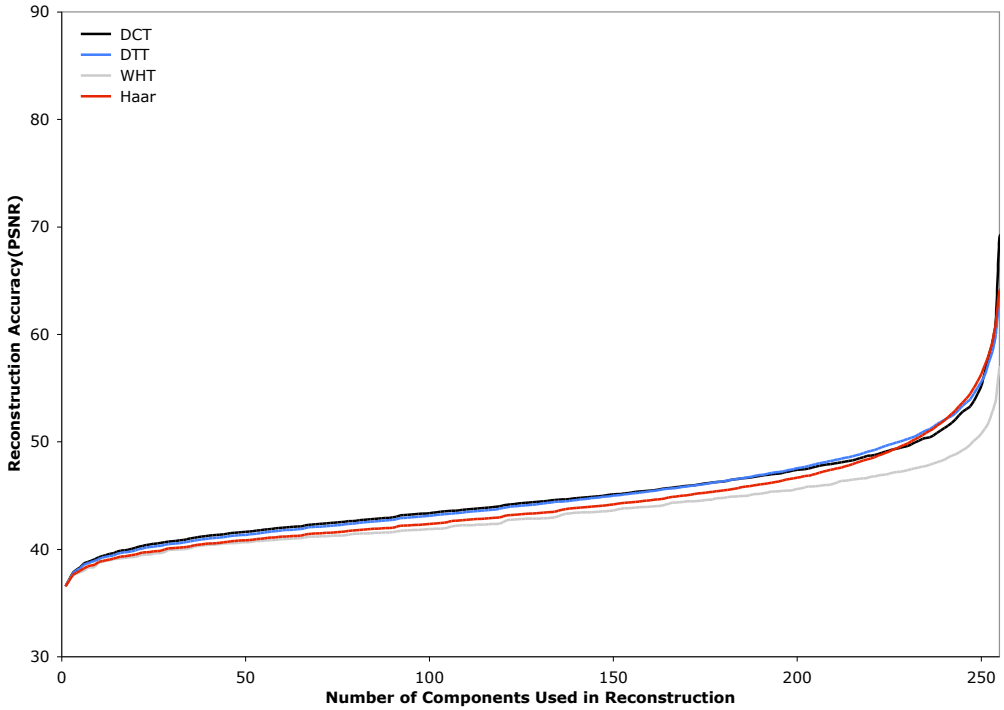


Figure 5.6 Reconstruction accuracy when processing the 'Goldhill' image.

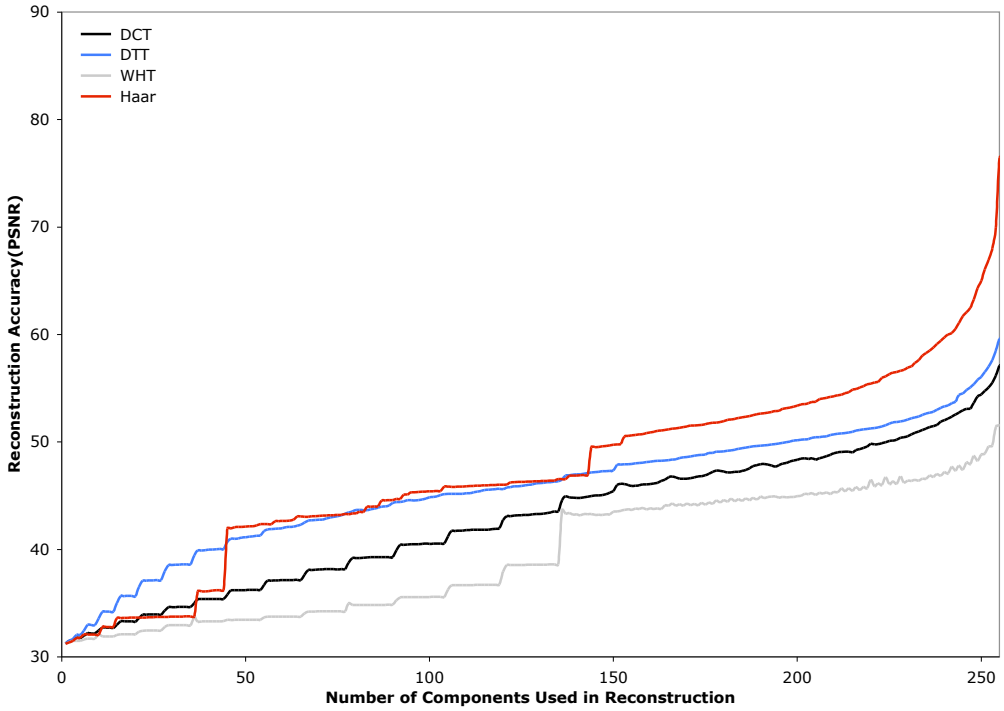


Figure 5.7 Reconstruction accuracy when processing the 'Pixel Ruler' image.

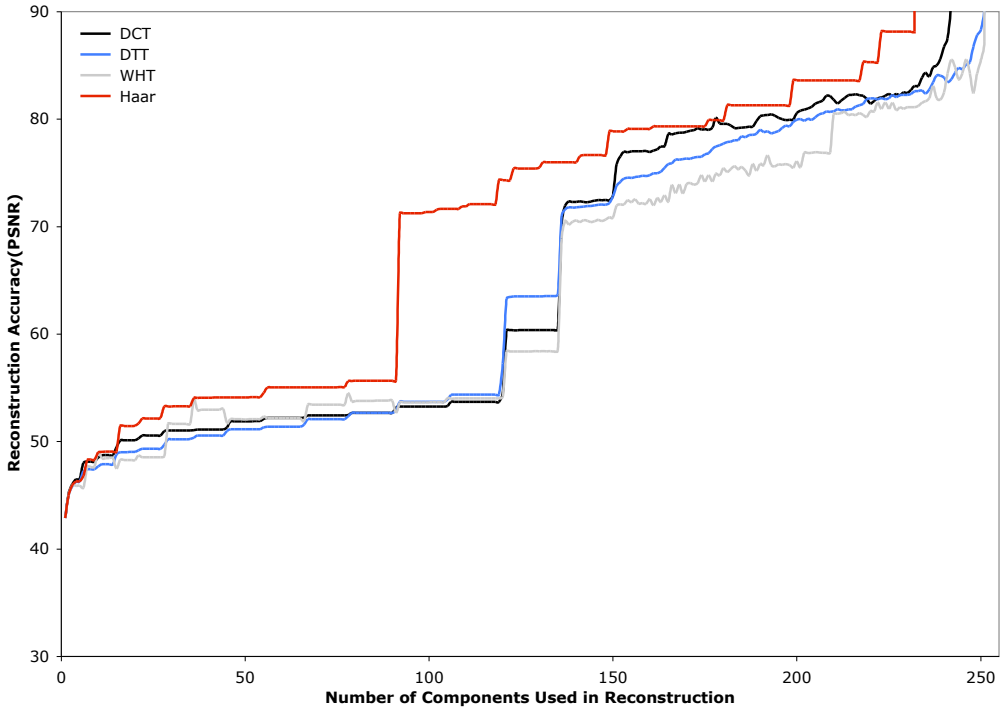


Figure 5.8 Reconstruction accuracy when processing the '21 Level Step Wedge' image.

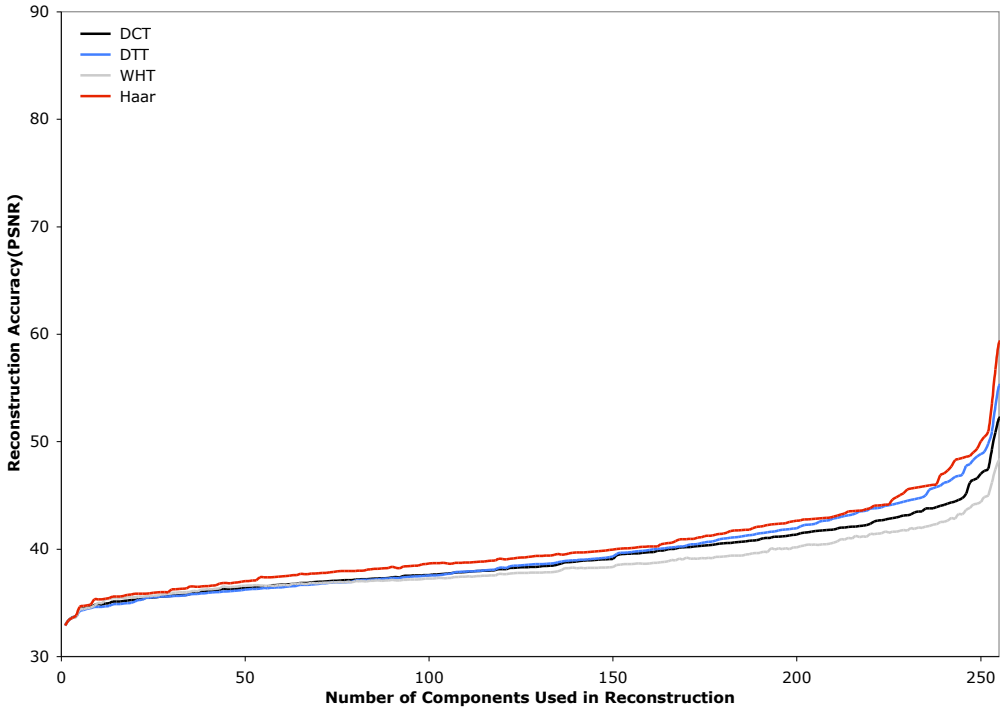


Figure 5.9 Reconstruction accuracy when processing the '256 Level Test Pattern' image.

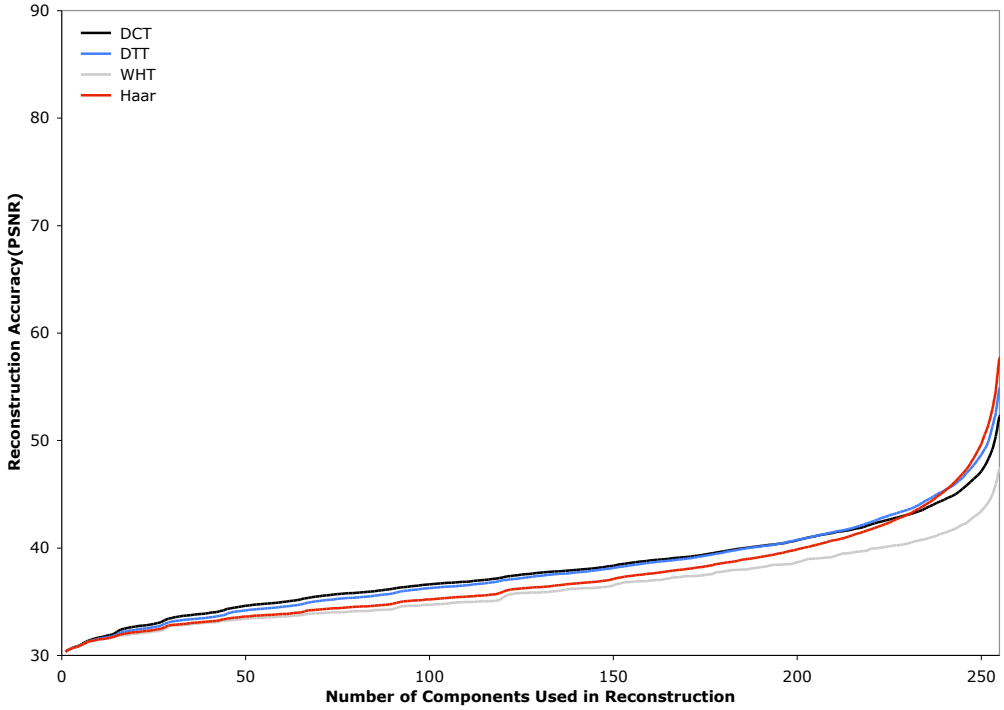


Figure 5.10 Reconstruction accuracy when processing the 'Texture Mosaic 2' image.

5.2.2 Summary of Results

Overall the simplest and most obvious fact that can be drawn is that the WHT has the lowest effective energy compactness of all the transforms we have studied. This manifests itself in a number of ways, most notably its abysmal performance compared to the remaining three transforms, regardless of the image being studied. While in most of images the WHT performed only marginally less effectively than the other transforms, occasionally even reaching the performance the other transforms the WHT never outperformed them by a reasonable margin. The best performance of the WHT was in the '21 Level Step Wedge' image, in which it occasionally outperformed the DCT and DTT. However given the amount of short term fluctuations in reconstruction accuracy produced by that image, the momentary leads given by the WHT were in no way convincing.

For the other three transforms however our task is slightly more difficult. In these there is a lot of fluctuation in the energy compactness provided by each transform. These fluctuations in performance are very clearly caused by variations in the types of image processed. The most obvious impact of different image types is the performance of the Haar transform. By examining those images in which the Haar transform performs well compared to other transforms that we are studying we can easily see the features that its is most attuned to. We can see that images with strong rectangular features tend to be favoured by the Haar transform. Generalising this to measurable image characteristics is somewhat more difficult however. The 'Pixel Ruler' demonstrates this measurably, as it is very predictable due to its grid nature, and the sharp lines result in a high SFM, causing difficulty for the other transforms.

The DCT and DTT both perform similarly for pictures that do not exhibit rapid intensity variations from one pixel to the next. These rapid variations correspond

with a high SFM value. For images that do exhibit such features (such as the 'Pixel Ruler') one can clearly see that the DTT outperforms the DCT. Looking at other images with large SFM values, such as the '256 Level Test Pattern' we can see that the DTT outperforms the DCT here as well, which seems to reinforce the observation that a high SFM leads to the DTT outperforming the DCT.

In images exhibiting low SAM values, the energy compactness of all of the transforms was reduced. This is due to the unpredictable nature of such images. Because of this lower energy compactness the all of the transforms demonstrated very similar performance. In cases where low SAM values were given, the DCT matches the performance of the DTT, or even exceeds it, even for high SFM values, as in 'Texture Mosaic #2'.

6 Discussion

In this chapter we discuss what we have found during the course of our study. A brief discussion of our results is given in Section 6.1. Following that, Section 6.2 will describe the limitations of our research. Areas of potential research that our project has found are then described in Section 6.3.

6.1 Performance and Classification

We can see from the results given in Section 5.2 that the performance of each of the transforms is strongly influenced by features in the image. Moreover each different transform is affected differently by these features. The results we have gathered allow us to make a number of generalisations about the performance of the transforms when processing different types of image.

- For natural images, or images with a low inter pixel variations (indicated by low SFM values) the DCT and DTT perform very well, outperforming both the WHT and the Haar transform.
- For images exhibiting high intensity variations between adjacent pixels, yet still being highly predictable the Haar transform provides the greatest energy compactness.
- In general the transforms perform similarly when processing images that have low SAM values.

We can classify images based on their SFM and SAM values. By using the results discussed in Section 5.2 we can associate each class of image with the transforms that provide the greatest energy compactness. This relationship is shown in Figure 6.1. In this chart we see that there are multiple choices for each situation. Unfortunately the measures we have used allow provide information about global features in each image. Unfortunately this does not allow us to quantify local features in an image that may be beneficial to specific transforms.

	SAM	
SFM	Low	High
Low		DCT or DTT
High	DCT, DTT, or Haar	DTT or Haar Transform

Figure 6.1 This small table shows which transforms demonstrate the highest energy compactness for certain combinations of image features.

The lack of any image exhibiting low SFM and SAM values means this table is not yet complete. Our problem in this situation is that any image that fitted in this

category would have only small variations in the intensity of neighbouring pixels, and would also need to present few predictable features. The only images that we would expect to match this description would need to be filled with low amplitude random noise.

6.2 Limitations

While not strictly problematic for our research there were a number of limitations we encountered. A number of the limitations of our approach subsequently provide avenues for further research and are discussed in Section 6.3.

6.2.1 Component Selection

One of the most significant limitations in our study was the way in which we selected transform components for use when reconstructing images. In all of our analyses we have selected the first N coefficients following the 'zig-zag' pattern[21]. In doing this we paid no heed to the effect a given component may have of the final reconstructed image. In most real world applications each component is scaled according to its relevance in the final image, and then thresholded. Only components that have a significant effect of the final image are then chosen. This approach provides a significant increase in the compression rates[22].

In order to accurately represent real world compression rates we would need to implement such a system, as suggested in Sections 6.3.1 and 6.3.2. We chose not to implement such a system, as our time constraints would not have allowed time for the studies required to develop quantisation tables or, as is discussed in Section 6.2.2, to perform the required quality studies.

6.2.2 Performance Analysis

Leading on from the problems inherent in our current component selection model we would ideally improve our performance measures. As it was our analysis was based on a raw comparison of signal to noise. As it has been stated, such measures fail to account for the human visual systems sensitivity (or lack of) to certain types of artifact in images[2, 9, 21].

To perform any reliable study accounting for human perception is very expensive and time consuming[2], and because of our time constraints such a study was infeasible. A more thorough study would need to perform some form of perceptive quality test in order to be able to accurately judge overall performance.

The other problem with our study was the absence of a full compression system. Ideally our analysis would have been able to perform the entire compression system, using either a custom pipeline, or perhaps a modified version of a well known standard, such as JPEG or some other similarly designed system.

6.2.3 Image Types

Our study was limited to just 8-bit grayscale images. While this limitation does not effect our ability to analyse the energy compactness of each transform, the data being transformed is comparable only to transformation in common RGB colour space. Many 'real world' systems, such as the JPEG and MPEG standards, are capable of compressing images in a number of different colour spaces. Given the differing structure present in different colour spaces, different compression results might be found.

6.3 Future Work

Our research has accomplished all of our stated goals. However, there is still a large quantity of research that can be performed in this domain. In this section we will discuss these possible fields of study, and describe why we feel that their output would be useful.

6.3.1 Improved Component Selection

In this report we focused exclusively on component selection through applying a cap on the maximum number of components used to reconstruct an image. This system has the advantage of being very simple, while still providing a good level of energy compaction. However there are more effective, though more complex, techniques in use.

The quantisation system used by the JPEG standard scales each component in the output transform by the corresponding value in a 'quantisation table'. If the resultant value is below a certain threshold then that component would not be transmitted[8]. This results in only transmission of those components that would actually have a significant impact on the reconstructed image.

Due to its use in a number of compression standards the quantisation tables for the DCT have been well analysed[20, 21]. For other transforms, such as the WHT, DTT and Haar transform very little research has been performed. Development of quantisation tables tailored to each of these transforms could make it possible to compare the actual compression rates of each transform.

6.3.2 Component Ordering

In our study we have used the 'zigzag' path used by the JPEG standard to determine the order of importance for each component in the transformed image. This approach has the advantage of giving the highest precedence to the lowest ordered component, and for many transforms of this type the information gained from these low order components represents the most significant details in the reconstructed image.

There are other mechanisms for ordering the transform components. If we look closely at the transformed Lenna images in Figure 4.2 we can see that the WHT and Haar Transform exhibit strong rectilinear features. It might be possible to use features such as these to improve the performance of transform that exhibit them. In the case of the Haar transform it becomes possible to reconstruct correctly scaled images for instance.

6.3.3 Image Domains and Structure

All of the analysis that we have performed has been on the intensity values of grayscale images. When performing compression on colour images we could continue to perform the compression this way, treating each of the red, green, and blue colour channels in the same way. However, when processing colour images there are a number of different ways in which the different colours can be represented, such as Hue-Saturation-Value, CIE[1], and YUV[2].

These different ways of representing coloured images may allow improved compression rates, as the different channels may exhibit features that are more readily

[1] The colour system defined by the Commision Internationale de L'Eclairage.

[2] Where the 'Y' component represents the luminance, and the 'U' and 'V' components represent chrominance (or base colour).

compressed. Moreover the different structure of each colour space may lead to the possibility of using different transforms on each channel. In RGB[3] colour space this has less relevance as each channel effectively has the same structure.

Such a study should also analyse the effect of different structures in the image. We have already noted that our current measures do not recognise local image features that may provide advantages to certain transforms.

6.3.4 'Real World' Analysis

Of course, the most obvious extension to this study would be the combination of all the other improvements we have discussed and the subsequent compression of the image.

[3]Red, Green, Blue

7 Conclusion

Two principle objectives were stated for this project. The first of these was to performed a detailed study on the energy compactness of a number of orthogonal image basis functions. The results of this study have been described in detail elsewhere in this report, but we will summarise them here.

Of all the transforms that we have analysed the WHT demonstrated the least energy compactness. It was found that significantly more components were needed from the WHT in order to provide the same reconstruction accuracy as the other basis functions. A more complex set of relationships was found when analysing the DCT, DTT, and Haar transforms.

We found that in general the Haar transform and the DTT would both provide higher energy compactness than the DCT in images with significant inter-pixel variations. In such images the DTT and Haar transform performed similarly, although in more predictable images the Haar transform could produce 'spikes' in reconstructions that could dramatically improve its performance.

For images containing natural scenes however, the DCT and DTT typically provide the highest energy compactness. In such images it was not possible to determine which of the two transform was the better, as the performance of each transform was similar. For specific details refer to our results and discussion in Sections 5.2.2 and 6.1.

Our second goal was to find some way of using global image features to estimate the most effective transform for a given image. To do this we studied the relationship of two image feature measures, the spatial frequency measure, and the spectral activity measure. While these measures could be used to find image features that affected the energy compactness of the transforms, we found that there was still a need to analyse local features of an image.

This research has shown that the now ubiquitous DCT is not necessarily the most effective image transform for compression. We have demonstrated that other transforms, notably the DTT, offer performance that rivals that of the DCT, and for certain classes of image can provide significant improvements.

BIBLIOGRAPHY

[1] N. Ahmed, T. Neterajan, and K. R. Rao. Discrete cosine transform. *IEEE Trans. on Computers*, 23:90–93, 1974.

[2] R. Algazi, M. Miyahara, and K. Kotani. Objective picture quality scale (pqs) for image coding. In *Proceedings of the Society for Information Display International Symposium*, volume XXIII, pages 859–862, 1992.

[3] Marc Antonini, Michel Barlaud, Pierre Mathieu, and Ingrid Daubechies. Image coding using wavelet transform. *IEEE Transactions on Image Processing*, 2(1):205–220, April 1992.

[4] D. Chen and A. C. Bovik. Visual pattern image coding. *IEEE Transaction on Communications*, 38(12):2137–2146, 1990.

[5] C. A. Christopoulos, T. Ebrahimi, and A. N. Skodras. Jpeg2000: the new still picture compression standard. In *Proceedings of the 2000 ACM workshops on Multimedia*, pages 45–49. ACM Press, 2000.

[6] I. Daubechies. *Ten Lectures on Wavelets*. CBMS-NSF Reg. Conf. Series in Applied Math. SIAM, 1992.

[7] A. Eskicioglu and P. S. Fisher. Image quality measures and their performance. In *IEEE Transactions on Communications*, pages 2959–2965, 1995.

[8] J. D. Gibson, T. Berger, T. Lookabaugh, D. Lindbergh, and R. L. Baker. *Digital Compression for Multimedia*. Morgan Kaufmann Publishers, Inc., 1998.

[9] S. Grgic, M. Mrak, and M. Grgic. Comparison of jpeg image coders. In *Proceedings of the 3rd International Symposium on Video Processing and Multimedia Communications*, pages 197–207, 1999.

[10] D. A. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101, 1952.

[11] O. Hunt and R. Mukundan. A comparison of discrete orthogonal basis functions for image compression. In *Image and Vision Computing New Zealand*, 2004.

[12] A. K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, 1998.

[13] N. Memon and X. Wu. Recent developments in context-based predictive techniques for lossless image compression. *The Computer Journal*, 40(2/3), 1997.

[14] R. Mukundan. Improving image reconstruction accuracy using discrete orthonormal moments. In *CISST 03 International Conference*, pages 287–293, 2003.

[15] R. Mukundan, S. H. Ong, and P. A. Lee. Discrete orthogonal moment features using chebyshev polynomials. In *Proc. of Intnl. Conf. on Image and Vision Computing*, pages 20–25, 2000.

[16] R. Mukundan, S. H. Ong, and P. A. Lee. Discrete vs. continuous orthogonal moments for image analysis. In *CISST 01 International Conference*, pages 23–29, 2001.

[17] R. Mukundan, S. H. Ong, and P. A. Lee. Image analysis by tchebichef moments. *IEEE Transactions on Image Processing*, 10(9):1357–1364, 2001.

[18] A.N. Netravali and J.O Limb. Picture coding: A review. *Proceedings of the IEEE*, 68:366–406, 1962.

[19] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423 and 623–656, July and October 1948.

[20] L. Teixeira and M. Martins. Video compression: The mpeg standards, 1996.

[21] G. K. Wallace. The jpeg still picture compression standard. *Commun. ACM*, 34(4):30–44, 1991.

[22] A. Watson. Image compression using the discrete cosine transform, 1994.

[23] Weinberger, Seroussi, and Sapiro. The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS. *IEEETIP: IEEE Transactions on Image Processing*, 9, 2000.

[24] Marcelo J. Weinberger, Gadiel Seroussi, and G. Sapiro. FROM LOCO-I TO THE JPEG-LS STANDARD. pages 68–72.

[25] T. A. Welch. A technique for high performance data compression. *IEEE Computer*, 17(6):8–20, June 1984.

[26] I. H. Witten, T. C. Bell, and A. Moffat. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann Publishers, Inc., 1999.

[27] X. Wu. An algorithmic study on lossless image compression. In *Data Compression Conference*, pages 150–159, 1996.

[28] X. Wu and N. Memon. Context-based, adaptive, lossless image coding (calic). *IEEE Transactions on Communications*, 45(4):437–444, 1997.

[29] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, IT-23(3):337–343, May 1977.

[30] J. Ziv and A. Lempel. Compression of individual sequences via variable rate coding. *IEEE Transactions on Information Theory*, IT-24(5):530–536, September 1978.

Publication of Work

A summary of this report has been published in the proceedings of the 'Image and Vision Computing New Zealand' conference, 2004[11]. The published work contains a summary of the research performed for this report. We discuss the computational aspects of the transformation process, and introduce the four transforms we have studied. Finally the paper briefly exams the performance of each transform over a pair of test images, demonstrating how different image properties effect the performance of each transform.

An online version of the published paper can be found at `http://studweb.cosc.canterbury.ac.nz/~ojh16/hunt04.pdf`.